

TDLPN: Transductive Dual Label Propagation Network for Few-shot Learning

Ye Li, Guangsheng Li

Abstract—The reliance on large labeled datasets for training deep Convolutional Neural Networks (CNNs) restricts their use in scenarios where labeled data are limited. In response, few-shot learning (FSL) enables knowledge transfer and the learning of new categories using very few labeled samples. This paper proposes a Transductive Dual Label Propagation Network (TDLPN) to address label scarcity and the propagation of label information. The TDLPN combines a Label Propagation Algorithm (LPA) with a Graph Neural Network (GNN). The LPA captures global relationships, while the GNN aggregates local neighborhood information and introduces momentum coefficients to enhance the label propagation process. Furthermore, a dot product attention method is used to build an attention-weighted graph that accurately depicts the relationships between input samples, and label predictions are derived from these data. According to the experimental findings, the TDLPN’s accuracy increases by 2.7% and 4.46%, respectively, on the miniImageNet dataset’s 5-way 5-shot and 5-way 1-shot tasks. Furthermore, this paper explores experimental methods optimized for single-shot label propagation on the miniImageNet, tieredImageNet, and CUB-200-2011 datasets, showcasing the improvement in few-shot classification performance and confirming that the TDLPN model exhibits strong generalization ability across diverse few-shot classification tasks.

Index Terms—Transductive learning, Few-shot learning, Label propagation, Graph neural networks, Attention-weighted graphs.

I. INTRODUCTION

IN numerous computer vision applications, including image classification [1] [2], semantic segmentation [3] [4], object detection [5] [6], and image caption generation [7], deep CNNs have demonstrated impressive success. But CNN training usually requires a lot of labeled data, like hundreds of samples for each category. For some rare categories, data collection and labeling are practically impossible, and this process is frequently both expensive and time-consuming. This issue has grown to be a significant barrier preventing deep learning methods from being widely used in practical applications. As a solution, FSL has received extensive research attention. Few-shot learning involves using a small number of labeled samples (e.g., only 1-5 samples per category) to achieve effective recognition of new categories by transferring knowledge from the base category with abundant samples. These methods typically leverage metric relationships between labeled and unlabeled data for image

classification. For instance, twin networks use weighted distances to determine the distances between positive and negative samples, enabling training to compare similarities [8]; matching networks learn the relationships between labeled and unlabeled sample sets using shared weighted metrics [9]; and prototype networks use distance computation in the metric space to sort samples [10], showing that these methods work well for classifying with few samples.

Many research groups are currently using FSL methods. Due to their superior performance in direct push learning settings, GNN-based few-shot learning techniques are becoming more and more popular. Because GNN can rapidly collect information through the graph structures they construct using a small number of support set and query set instances, few-shot learning works well with them. This is achieved by passing messages, which rapidly gather information. These approaches use GNN as the core module for label propagation [11] and leverage the graph structure for node label prediction [12] or edge label prediction [13]. Stated otherwise, the GNN serves as a classifier in this context, converting the feature-embedding network’s output into category labels. The GNN parameters and feature embedding network parameters are considered of as two parts of the same model that are co-learned and optimized together in an outer loop.

The Transductive Propagation Network (TPN) [14] faces several issues when dealing with changes in graph structure. It is highly dependent on graph structure, and may not effectively propagate label information when the topology of the graph changes. The TPN depends on the label information of the labeled nodes to propagate in sparse labeling scenarios; the classification accuracy may be decreased if the label information of unlabeled nodes is not fully utilized. This is primarily due to the fact that the one-time label propagation mechanism of TPN fails to fully leverage the migration potential of the graph structure across tasks, thereby limiting its generalization ability. To address these issues, this study proposes a new few-shot classification method, the Transductive Dual Label Propagation Network (TDLPN). A graph is first constructed to represent the node and edge features of the input samples and to reveal the relationship between all input samples. To get their respective predictions, this graph is given into the label propagation algorithm and the graph neural network. This approach fully utilizes the graph structure information by integrating the Graph Attention Network (GAT) and the Label Propagation Algorithm (LPA), which improves the accuracy and efficiency of few-shot classification. In tests, the TDLPN model improves accuracy by 4.46% and 2.7% on the 5-way 1-shot and 5-way 5-shot tasks on the miniImageNet dataset.

The main contributions of this paper include:

- 1) A TDLPN few-shot classification method that combines GNN and LPA is proposed. The method uses

Manuscript was received November 18, 2024; revised April 12, 2025.

Ye Li is a senior engineer of the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, China (e-mail: liye@usst.edu.cn).

Guangsheng Li is a postgraduate student of the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, China (Corresponding author, e-mail: 223330764@st.usst.edu.cn).

LPA and GAT to gather global information about the sample points and local neighborhood information about each node, respectively. In this way, the approach can completely utilize the label information of both labeled and unlabeled nodes, as well as capture the global structure of the data and the local relationships.

- 2) The momentum coefficient is introduced to optimize the label propagation algorithm. As a result, the momentum term helps make the process of updating labels smoother and less prone to oscillations and instability, thereby speeding up the convergence of label propagation.
- 3) An attention-weighted graph is constructed using the attention weights obtained from the dot product attention mechanism as the edges of the graph. Since attention weights accurately capture the interrelationships between the input samples, the labeling information is propagated more effectively.

II. RELATED WORK

A. Metric learning with few-shot learning

In the context of few-shot learning, metric learning methods emphasize the importance of learning similarity classifiers over the feature space [15]. They achieve this by employing a cross-task neural network backbone design and concentrating on learning high-quality, transferable features. Using learnt embeddings from a limited number of labeled samples (the support set), the Matching Network (MN) uses an end-to-end trainable nearest neighbor method to predict the class of unlabeled data (the query set). By creating a prototype representation for every category to facilitate classification, the Prototypical Network (PN) expands on this idea. Relation Networks use a basic neural network to learn a nonlinear distance metric rather than depending on conventional fixed linear distance metrics like cosine or Euclidean distance. End-to-end networks are typically trained with small amounts of training data in these methods. The idea is that the features learned during training can accurately represent the new test categories. By teaching a neural network to use a nonlinear distance metric and then combining it with Euclidean distance to calculate similarity and build the adjacency matrix, it is feasible to obtain a more accurate image of how similar two sets of data are, particularly in high-dimensional areas.

B. Graph-based learning with few-shot learning

In the research field of FSL, graph-based methods are often regarded as a special branch of metric learning. The reason for this is that most of these methods depend on Radial Basis Function (RBF)-based adjacency matrices to create graphs that facilitate for labels or features to spread. For instance, by creating affinity matrices between the support set and unlabeled data, Satorras et al. [16] were able to effectively propagate labels. In order to regularize the feature representation. To regularize the feature representation, wDAEGNN [17] used graph neural networks (GNNs) to generate classification weights and combined them with denoising autoencoders (DAEs). In addition, embedding propagation methods propagate not only the labels but also the embeddings themselves to reduce intra-class distance and thus improve classification

accuracy [18]. Graph Convolutional Networks (GCNs) have been utilized to instantiate set-to-set functions, which have also been employed for embedding adaptation [19].

C. Transductive learning and semi-supervised few-shot learning

One of the most common paradigms in few-shot learning is the inductive setting. In this mode, the learning or fine-tuning process of the model is limited to determining the labels of query samples based only on the samples in the support set. In contrast, the direct inference setting diverges from the inductive approach by enabling the model to access all query samples for classification, including those devoid of explicit labels. In certain studies, query samples and entropy minimization techniques are fine-tuned using query samples to enhance the model's certainty of prediction. In addition, techniques such as label propagation and embedding propagation are used for representation learning, which is similar to their application in meta-learning.

In semi-supervised few-shot learning, unlabeled data is supplied alongside a labeled support set. Although the distribution of this data is thought to be comparable to that of the target category, it might also include some irrelevant samples. The LST [20] study employs self-labeling and soft attention to process unlabeled samples concurrently and intermittently. Subsequently, labeled and self-labeled data are amalgamated to fine-tune the model. The LST method is similar to the approach adopted by Ren et al. [21], who utilized K-means iterations initiated by a prototype network to update category prototypes and mitigate the impact of samples that might not belong to the target category during the processing of unlabeled data. Simon et al. [22] also employed unlabeled samples and soft label propagation techniques. Additionally, Saito et al. [23] explored the challenge of semi-supervised, few-shot domain adaptation. In other studies [24], graph neural networks were applied to semi-supervised FSL settings to facilitate information sharing between labeled and unlabeled samples, with a graph construction network used in TPN to predict task-specific graphs for propagating labels in a semi-supervised FSL task. Finally, Liu et al. [25] pointed out the bias between the prototype representation and the ideal representation and proposed a simple strategy based on intra-class and inter-class assumptions to correct this bias.

III. METHODOLOGY

A. Problem definition

Few-shot classification focuses on building classifiers when only limited training samples are available for each class. A typical few-shot task T includes a support set S (labeled input-label examples) and a query set Q (unlabeled samples used for evaluation). When the support set S contains K labeled samples for each of N different categories, this situation is called an N-way K-shot classification problem.

Meta-learning has recently emerged as a widely used approach for addressing few-shot classification challenges. Few-shot learning can use the compact support set of a task alone to train classifiers to assign a class label to each sample in the query set. However, the small number of labeled support samples makes it impossible to train

models that accurately reflect differences between and within classes, which usually leads to poor classification performance. Meta-learning solves this problem by transferring relocatable knowledge from the explicit training set. This allows models to learn faster with few shots on the support set, leading to better classification of the query set.

In this paper, we adopt scenario training as an efficient method for meta-learning. The core idea of scenario training is to sample training tasks (i.e., scenarios) from a relatively large labeled training dataset that mimics the few-shot learning environment of the test task. Since the distribution of the training tasks is considered to be similar to that of the test task, the performance on the test task can be improved by ensuring that the model performs well on the training tasks.

Specifically, the training and testing tasks in scenario training are constructed as N-way K-shot problems:

$$\Gamma = S \cup Q \quad (1)$$

where: Γ denotes the task, $S = \{(x_i, y_i)\}_{i=1}^{N \times K}$ and $Q = \{(x_i, y_i)\}_{i=N \times K+1}^{N \times K+T}$ are the support and validation sets, respectively, N denotes the number of categories, K indicates the count of instances per category, T the number of query samples. $x_i, y_i \in \{C_1, \dots, C_N\} \subset C$, are the i -th input data and its label, respectively, and C is the set of all classes in the training or test dataset. While train and test tasks are sampled from the same distribution, their label sets do not overlap $C_{\text{train}} \cap C_{\text{test}} = \emptyset$. The support set S in each scenario acts as the labeled training set on which the model is trained with the aim of minimizing the prediction loss on the validation set Q . The model is trained on the training set S and the validation set Q . This training process is repeated until the model converges. When some of the $N \times K$ support instances lack labels, the task is termed semi-supervised few-shot learning.

B. Models

TDLPN contains four key components, as shown in Figure 1. First, by employing feature embedding, the model becomes more effective in capturing and representing input information. Secondly, the graph construction module effectively utilizes the data's flow structure by generating node and edge features, ultimately generating an attention-weighted graph through a point-by-attention mechanism. After that, the graph construction is applied to both GAT and LPA to propagate labels from the support set S to the query set Q . Finally, a bilevel optimization [26] algorithm is used in the loss generation phase to compute the cross-entropy loss between the propagated labels and the true labels on the query set Q and to optimize all the parameters in the joint training framework. This integrated approach not only improves the accuracy of the model in dealing with complex data structures but also enhances its generalization ability in few-shot learning tasks.

1) *Feature embedding*: In this module, Conv4 and ResNet12 networks are adopted in this paper to extract the input x_i features, respectively:

$$v_i = f_{\varphi}(x_i, \varphi) \quad (2)$$

where φ denotes the parameters of the network. The node features in the support set and query set are extracted by the same embedding.

2) *Diagram construction*: In this research, we suggest a trainable approach to generate edge features instead of using a fixed similarity measure such as K-nearest neighbors with a Gaussian kernel. According to research [27], the self-attention mechanism uses the attention value to determine how different parts of the input are related, even when the distance is taken into account. Inspired by this, to efficiently build neighborhood graphs in the framework of meta-learning, this paper constructs graphs based on the support and query sets. In this module, CNN is used to generate a feature representation unique to each sample. The edge weights of each sample are computed through a multiplicative dot product attention mechanism, which is uniquely determined for each sample and obtained through contextual training. This method can adapt to different task requirements and optimize the performance of few-shot learning. With the introduction of the multiplicative dot product attention mechanism, edge weights are computed by first applying a linear transformation to each node:

$$h_i = W x_i \quad (3)$$

where W is the linear transformation matrix and x_i is the eigen representation of node i .

Also, using the multi-head mechanism, unions are performed in h different representation subspaces to minimize random errors. Compute the multi-head dot product attention score:

$$e_{ij}^{(k)} = \frac{(h_i^{(k)} \cdot h_j^{(k)})}{\sqrt{d_k}} \quad (4)$$

where d_k indicates the feature dimension of the k attention head, $h_i^{(k)}$ denotes the feature performance of node i at the k attention head, and \cdot denotes the dot product.

Normalize the attention scores to obtain the edge weights of each attention head, splice the edge weights of all attention heads, and obtain the final edge weight matrix W through a linear transformation:

$$\alpha_{ij}^{(k)} = \text{softmax}(e_{ij}^{(k)}) = \frac{\exp(e_{ij}^{(k)})}{\sum_l \exp(e_{il}^{(k)})} \quad (5)$$

$$W_{ij} = \text{Concat}(\alpha_{ij}^{(1)}, \alpha_{ij}^{(2)}, \dots, \alpha_{ij}^{(h)}) W^0$$

where W^0 is the linear transformation matrix used to integrate the results of the multi-head attention.

In Figure 2, the suggested graph creation module is displayed. This construction no longer relies solely on the traditional method of matching and comparing between the support set and the query set. Instead, it reveals the deep relationships between the entire input data through an output graph that depicts in depth the comprehensive connections between the support set and the samples in the query set. Such a graph structure, as important as prior information, utilizes the correlation between the samples to effectively facilitate the feature extraction and aggregation process, providing a more global perspective and support for subsequent training.

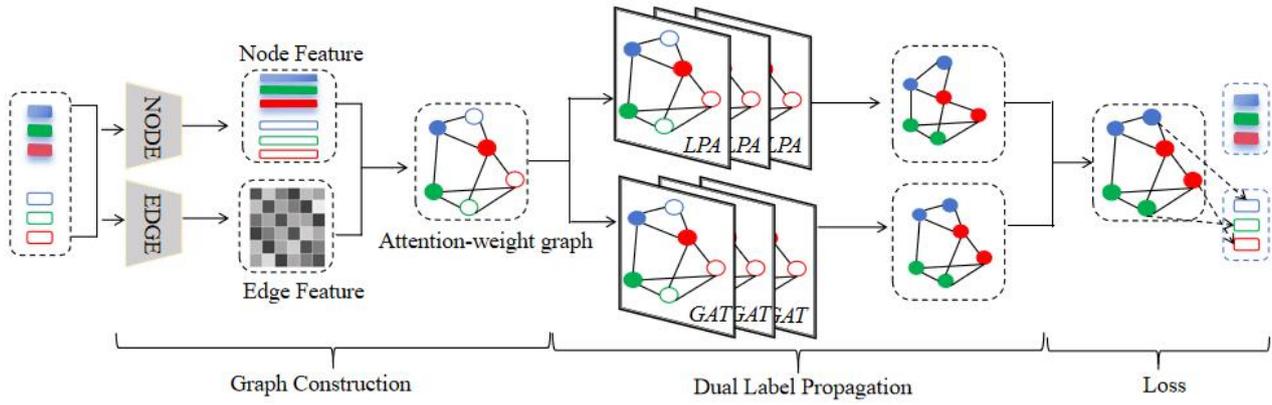


Fig. 1. TDLPN model architecture diagram

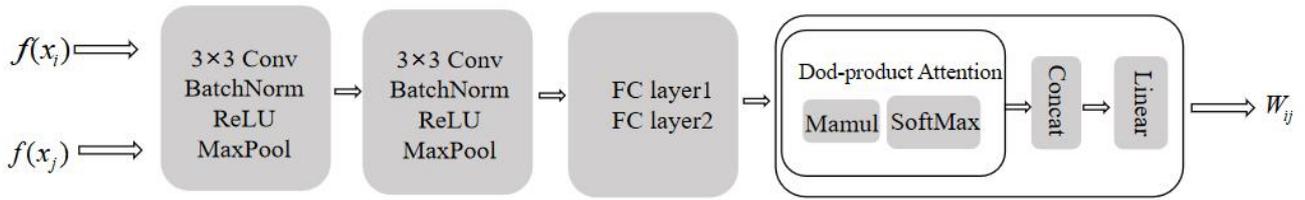


Fig. 2. Diagram construction module

In the training phase, the scenario training method is used to construct the graphs, where the size of the graphs flexibly varies according to the number of randomly selected samples in the support set and query set. The graphs created for each scenario are separate from each other. This method effectively reduces the amount of computation needed to process all the samples, making the graph construction process more targeted and efficient.

3) *Double labeling propagation*: A semi-supervised learning strategy is used after the graph structure of the input image data is learned. This is done with a graph attention network and a label propagation algorithm that sends label information between labeled and unlabeled connected neighbors. This approach enables the model to learn rapidly even in the absence of many labels by using structural links between unlabeled data. Additionally, it enhances the model's capacity to adjust to fresh data.

1) Graph Attention Networks

A graph-structured graph attention network approach is used to predict the labels of the query set Q . Each node in the network is given a varied weight depending on how important its neighbors are, according to a self-attention process. With weights established by the attention mechanism, each node's features are updated as a weighted sum of the features of its nearby nodes. The nodes are then subjected to a self-attention operation in order to acquire a shared attention mechanism for the purpose of computing the attention coefficients:

$$e_{ij} = a(Wh_i, Wh_j), j \in N_i \quad (6)$$

where a is a learnable attention mechanism, W is the feature transformation weight matrix, h_i and h_j are the feature vectors of the node and node respectively, $j \in N_i$, j is the neighbor node of the node i in the graph. All nodes j and

e_{ij} are normalized using the softmax function to get the final attention weights, which make it simple to compare the coefficients across various nodes:

$$h'_i = \sigma \left(\sum_{j \in N_i} a_{ij} Wh_j \right) \quad (7)$$

In order to improve the stability of the self-attention learning process, this paper employs a multi-head attention mechanism. It does this by running K separate attention mechanisms at the same time, each of which performs the same transformation, and then combining their output features to make the final output feature representation:

$$h'_i = \sigma \left(\frac{1}{K} \sum_{k=1}^k \sum_{j \in N_i} a_{ij}^k W^k h_j \right) \quad (8)$$

where W^k is the weight matrix of the corresponding input linear transformation, and a_{ij}^k is the normalized attention coefficient calculated by the k attention mechanism.

2) Label propagation

The labels of the query set Q are predicted using a graph-structured label propagation technique. Define F as a set of $(N \times K + T) \times N$ matrices containing non-negative elements. In the labeling matrix Y , the functional form of Y is:

$$Y = \begin{cases} 1 & x_i \in S, y_i = j \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

The unknown labels of $S \cup U$ instances in the iterative graph structure update serve as the basis for label propagation. The original formula for label propagation is:

$$F = (I - \alpha S)^{-1} Y \quad (10)$$

The formula in the iterative label propagation process is:

$$F(t+1) = \alpha SF(t) + (1-\alpha)Y \quad (11)$$

Add the momentum coefficient μ to equation (12) to get the formula:

$$F(t+1) = \mu F(t) + (I-\mu)(\alpha SF(t) + (I-\alpha)Y) \quad (12)$$

where I is a unit matrix, $F(t)$ representing the predicted label distribution at time t , S denotes a normalization matrix over W , $\alpha \in (0, 1)$ is a trade-off factor to balance the importance of the information in the propagation process and the original label information, and μ is a momentum factor. With the introduction of the momentum coefficient μ , the new label propagation algorithm can converge to a steady state more quickly, thus improving the efficiency and accuracy of label propagation.

C. Model Optimization

1) *Model Loss*: Label propagation in few-shot learning can leverage unlabeled data to improve classification performance by conveying label information through the graph structure, which helps to learn more generalized feature representations. However, it relies on the quality of graph construction and is sensitive to noisy labels. Graph Attention Networks (GATs) are able to dynamically focus on important neighboring nodes through an attention mechanism to enhance feature representation. Thus, the use of unlabeled data improves generalization ability through label propagation while capturing complex node relationships through GAT, resulting in better performance in few-shot learning tasks. Model combination can be realized by assigning higher weights to the corresponding model predictions. The combined prediction formula is:

$$y = \frac{\exp(\theta_1)}{\sum_{i=1}^2 \exp(\theta_i)} y^B + \frac{\exp(\theta_2)}{\sum_{i=1}^2 \exp(\theta_i)} y^G \quad (13)$$

where y^B is the prediction of the label propagation, y^G is the prediction of the graph attention network, θ_1 and θ_2 are learnable weights to control the contribution of the two models in the final prediction. θ_1 and θ_2 can be chosen based on their performance on the validation set, and more specifically, this objective can be expressed as the following two-layer optimization problem:

$$\begin{aligned} \min_{\theta_1, \theta_2} \mathcal{L}_{\text{val}}(\varphi_B^*(\theta_1, \theta_2), \varphi_G^*(\theta_1, \theta_2), \theta_1, \theta_2) \text{ s.t. } \varphi_B^*, \varphi_G^* \\ = \arg \min_{\varphi_B, \varphi_G} \mathcal{L}_{\text{train}}(\varphi_B, \varphi_G, \theta_1, \theta_2) \end{aligned} \quad (14)$$

where \mathcal{L}_{val} and $\mathcal{L}_{\text{train}}$ represent the cross-entropy loss on the validation and training sets, respectively, and are used to measure the accuracy of the model in predicting labels on both datasets.

2) *Alternate optimization*: In order to reduce the computational cost and storage requirements, this thesis employs an alternating optimization strategy that first fixes θ_1 and θ_2 , by optimizing φ_B and φ_G to minimize the loss on the training set. Then fixes φ_B and φ_G , and optimizes θ_1 and θ_2 to achieve the best results on the validation set. This alternating

approach effectively balances computational efficiency and resource consumption. In order to further accelerate the optimization process, a mechanism to dynamically adjust the learning rate is introduced so that the model automatically adjusts the step size according to the gradient changes in different iterations to achieve faster convergence.

In this paper, instead of computing φ_B^* and φ_G^* during the upper-level optimization process, we fix θ_1 and θ_2 and update the model parameters φ_B and φ_G in the t -step by:

$$\begin{aligned} \varphi_B^{t+1} &= \varphi_B^t - \alpha_B \nabla_{\varphi_B} \mathcal{L}_{\text{train}}(\varphi_B^t, \varphi_G^t, \theta_1, \theta_2) \\ \varphi_G^{t+1} &= \varphi_G^t - \alpha_G \nabla_{\varphi_G} \mathcal{L}_{\text{train}}(\varphi_B^t, \varphi_G^t, \theta_1, \theta_2) \end{aligned} \quad (15)$$

where φ_B^t and φ_G^t are the model parameters after updating t steps. α_B and α_G are the learning rates of φ_B and φ_G .

To further speed up the optimization process, this paper introduces a one-step approximation method to compute the upper layer parameters θ_1 and θ_2 gradient updates:

$$\begin{aligned} \theta_1^{k+1} &= \theta_1^k - \alpha_\theta \nabla_{\theta_1} \mathcal{L}_{\text{val}}(\varphi_B^T, \varphi_G^T, \theta_1^k, \theta_2^k) \\ \theta_2^{k+1} &= \theta_2^k - \alpha_\theta \nabla_{\theta_2} \mathcal{L}_{\text{val}}(\varphi_B^T, \varphi_G^T, \theta_1^k, \theta_2^k) \end{aligned} \quad (16)$$

where φ_B^T and φ_G^T denote the stopping gradient, and α_θ is the learning rate of θ_1 and θ_2 .

IV. EXPERIMENTS

A. Datasets

The experimental phase made use of three well-known FSL quasi-datasets: miniImageNet, tieredImageNet, and CUB-200-2011. Each of the 100 classes that make up miniImageNet's composition has 600 images. According to the criteria [28], the photos are separated into 20 test classes, 16 validation classes, and 64 training classes. A broader subset of ImageNet ILSVRC-12, TieredImageNet comprises 779,165 images in 608 categories, which are further subdivided into 160 test classes, 97 validation classes, and 351 training classes. With 11,778 photos of 200 bird species split into 100 training classes, 50 validation classes, and 50 test classes, CUB-200-2011 is a fine-grained categorization dataset in contrast to the other two. The fact that every image in the collection has been normalized to 84×84 size is notable.

B. Setup

Like many other CNN-based image recognition tasks, the FSL model requires a feature embedding network. The effectiveness of this network is greatly influenced by its backbone. Two popular backbones, Conv-4 and ResNet-12, were utilized in the trials to allow for a fair comparison with earlier methods. The final output feature dimension of the Conv-4 backbone, which consists of four convolutional blocks, is 64. The majority of the most advanced models employ the ResNet-12 backbone. With an output feature dimension of 640, it is composed of four residual blocks. The Adam optimizer is used in all trials, with an initial learning rate of 10^{-3} and α set to 0.99 for label propagation. The learning rate was halved for every 10,000 training sets in the miniImageNet and CUB-200-2011 datasets, and halved for every 25,000 training sets in the tieredImageNet dataset. Because tieredImageNet has more categories and

more samples in each category, it takes longer training cycles to accomplish the learning effect. For this reason, the learning rate is decreased at bigger intervals. Until the validation loss achieves an equilibrium state, the training process continues.

C. Results of the experiment

1) *Key results:* This paper presents a comparative analysis between the proposed approach and other state-of-the-art models. The include graph-based models like TPN [14], DPGN [13], BGNN [12], EGNN [24], HGNN [28], and non-graph-based approaches, including RelationNet [15], MatchingNet [9], ProtoNet [10], MAML [29], MetaGAN [30], SNAIL [31], Meta-Transfer [32], TapNet [33], Closer-Look [34], FEAT [35], E3BM [36], and MetaOptNet [37].

Tables I to III illustrate the effectiveness of the one-shot label propagation network (TPN), which has been broadly recognized in the field. From Table I, it can be seen that TPN achieves 59.46% and 75.65% accuracy in the 5-way 1-shot and 5-way 5-shot settings on the miniImageNet dataset, respectively. However, TPN still has drawbacks when handling sparsely labeled data, despite its strong performance in graph structure analysis. The TDLPN is proposed in this work as a solution to these issues. By facilitating the model's ability to adjust to modifications in the graph's structure, it raises classification accuracy.

On the same dataset, TDLPN improves its performance to 63.92% and 78.35% in the 5-way 1-shot and 5-way 5-shot settings, respectively, which are 4.46% and 2.7% higher compared to TPN. This significant improvement is due to the fact that the TDLPN's graph structure understanding and label propagation algorithms have been enhanced. This makes them more reliable when labels are scarce. After extensive analysis, it is evident that the enhanced model's greater ability to adapt to changes in graph structure is what caused the results to significantly improve. This makes the classification more accurate when labels are scarce. In particular, the model fully utilizes the GAT and the LPA by combining the graph's local and global information. The GAT aggregates the local neighbor information for each node and expands the sensory field through multi-layer stacking, whereas the LPA captures the global relationships between sample points. This design allows the model to learn and adapt better when dealing with complex graph structures, thus demonstrating higher accuracy and generalization ability in few-shot learning tasks.

In both 5-way 1-shot and 5-way 5-shot settings, the enhanced technique outperforms existing graph-based and non-graph-based few-shot methods in terms of accuracy on the miniImageNet dataset. Except for the method proposed in this paper, the method with the best results in both 1-shot and 5-shot is HGNN, with a result of 72.48%, which is still about 2% lower than the TDLPN in this paper. Furthermore, the approach suggested in this research continues to produce good results on the CUB-200-2011 dataset and the tieredImageNet dataset.

2) *Semi-supervised few-shot classification:* In this study, the miniImageNet dataset is divided into groups with different ratios of labeled and unlabeled samples. This is done to compare four methods: GNN [24], EGNN [24], TPN [14], and EGNN(T) [24]. The findings of the comparison, which

TABLE I
ACCURACY OF 5-WAY 1-SHOT AND 5-WAY 5-SHOT TASKS ON
MINIIMAGENET DATASET

Method	Backbone	5way-1shot	5way-5shot
MatchingNet [9]	Conv4	43.56±0.84	55.31±0.73
ProtoNet [10]	Conv4	49.42±0.78	68.20±0.66
MAML [29]	Conv4	48.70±1.84	55.31±0.73
TPN [14]	Conv4	55.51±0.86	69.86±0.65
DPGN [13]	Conv4	53.22±0.31	65.34±0.29
BGNN [12]	Conv4	52.35±0.42	67.35±0.35
EGNN [24]	Conv4	51.65±0.55	66.85±0.49
HGNN [28]	Conv4	55.63±0.20	72.48±0.16
TDLPN (ours)	Conv4	57.34±0.25	74.28±0.22
MetaGAN [30]	ResNet12	52.71±0.64	68.63±0.67
SNAIL [31]	ResNet12	55.71±0.99	68.88±0.92
Meta-Transfer [32]	ResNet12	61.20±1.80	75.53±0.80
TPN [14]	ResNet12	59.46±n/a	75.65±n/a
TapNet [33]	ResNet12	61.65±0.15	76.36±0.10
TDLPN (ours)	ResNet12	63.92±0.32	78.35±0.15

TABLE II
ACCURACY OF 5-WAY 1-SHOT AND 5-WAY 5-SHOT TASKS ON THE
TIEREDIMAGENET DATASET

Method	Backbone	5way-1shot	5way-5shot
MatchingNet [9]	Conv4	54.02±0.00	70.11±0.00
ProtoNet [10]	Conv4	50.89±0.21	69.26±0.18
MAML [29]	Conv4	51.67±1.81	70.30±0.08
TPN [14]	Conv4	57.53±0.96	72.85±0.74
DPGN [13]	Conv4	53.99±0.31	69.86±0.28
BGNN [12]	Conv4	49.41±0.43	65.27±0.35
EGNN [24]	Conv4	47.40±0.43	62.66±0.57
HGNN [28]	Conv4	56.05±0.21	72.82±0.18
TDLPN (ours)	Conv4	59.66±0.13	76.28±0.21
TPN [14]	ResNet12	59.91±0.94	73.30±0.75
TapNet [33]	ResNet12	63.08±0.15	80.26±0.12
Meta-Transfer [32]	ResNet12	65.62±1.80	80.61±0.90
MetaOptNet [37]	ResNet12	65.81±0.74	81.75±0.53
ProtoNet [10]	ResNet12	69.63±0.53	84.82±0.36
E3BM [36]	ResNet12	70.00±n/a	85.00±n/a
TDLPN (ours)	ResNet12	71.84±0.16	85.64±0.12

is conducted in an environment with little labeled data, are displayed in Figure 3. These methods have demonstrated significant efficacy or introduced novel learning mechanisms in previous studies and were thus chosen to show the variation in their performance with different ratios of labeled to unlabeled samples. The support samples in each category are divided into labeled and unlabeled parts according to the semi-supervised scenarios where the proportion of labeled samples is set to 0.2, 0.4, and 0.6.

By meticulously analyzing the similarity distribution among all samples, this study reveals the efficiency of information transfer in semi-supervised learning, especially when labeled data are scarce. During the experiments, it is first observed that the performance of all models generally decreases as the proportion of labeled samples decreases. This finding aligns with expectations, since less labeled information may make it difficult for classifiers to accurately establish boundaries between categories. However, TDLPN helps mitigate this downward trend by effectively extracting and utilizing the structured information embedded in unlabeled samples. In comparison to a number of alternative approaches, the findings shown in Figure 3 confirm that the approach suggested in this research can maintain high classification accuracy in resource-constrained circumstances. This not only demonstrates the robustness of the method but also highlights its potential application in the field of semi-

TABLE III
ACCURACY OF 5-WAY 1-SHOT AND 5-WAY 5-SHOT TASKS ON THE CUB-200-2011 DATASET

Method	Backbone	5way-1shot	5way-5shot
MatchingNet [9]	Conv4	61.16±0.89	72.86±0.70
ProtoNet [10]	Conv4	51.31±0.91	70.77±0.69
MAML [29]	Conv4	55.92±0.95	72.09±0.76
RelationNet [15]	Conv4	62.45±0.98	76.11±0.69
CloserLook [34]	Conv4	60.53±0.83	79.34±0.61
TDLPN (ours)	Conv4	65.35±0.56	81.47±0.51
FEAT [35]	ResNet12	68.87±0.22	82.90±0.15
TDLPN (ours)	ResNet12	76.54±0.46	88.06±0.39

supervised few-shot learning.

Specifically, compared with methods such as TPN and EGNN, TDLPN utilizes labeled support samples and can accurately establish the similarity distribution among all samples, which in turn facilitates the deep connection between labeled and query samples and enables the efficient transfer of information from labeled samples to query samples. The proposed method outperforms the existing few-shot semi-supervised methods, as shown in Figure 3. The findings also demonstrate that, despite a decline in the labeling ratio, it successfully capitalizes on the link between labeled and unlabeled data.

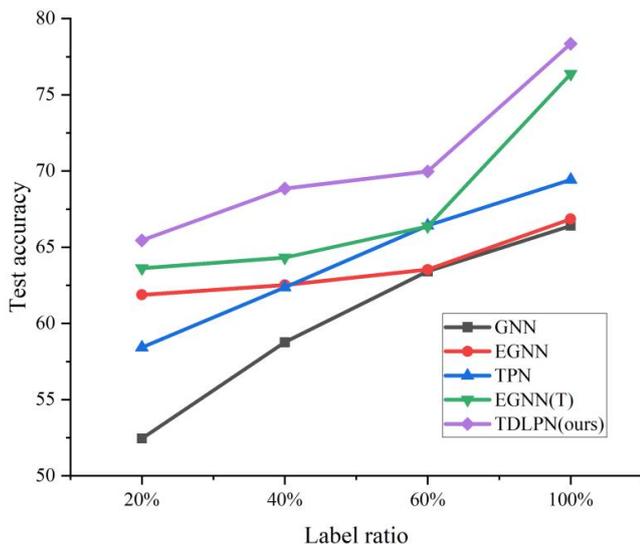


Fig. 3. MiniImageNet results in semi-supervised scenarios

3) *High-way few-shot classification*: In addition, the performance of TDLPN in high-category few-shot scenarios was evaluated on the miniImageNet dataset. As shown in Figure 4, TDLPN not only outperforms powerful graph-based methods, but also significantly outperforms non-graph methods such as MAML [29], RelationNet [15], and MetaOpt [37]. As the number of few-shot tasks increases, LPA captures global information about the graph, GAT can aggregate local information, and TDLPN continues to extract more detailed information for queries by combining both.

4) *Model complexity analysis*: Assessing a deep learning model’s complexity is one of the most crucial methods to determine how effective it is. This is usually measured by counting the number of floating-point operations (FLOPs), the number of parameters (Params), and the memory footprint (Memory). In this paper, we experimentally compare the complexity of GNN, DN4, DPGN, and TDLPN models.

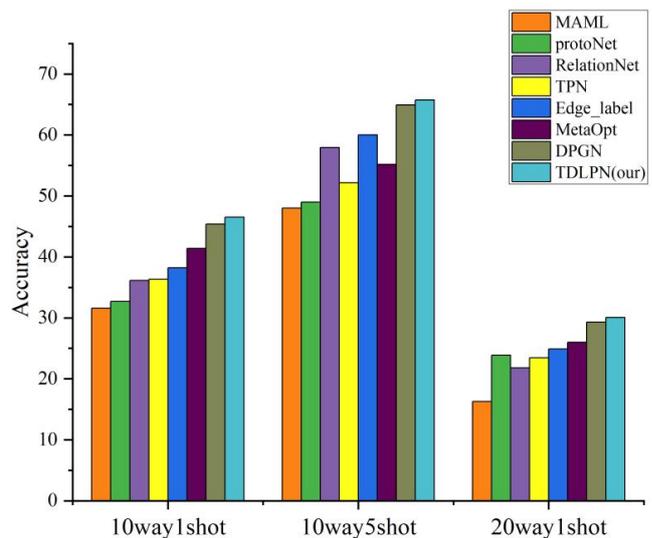


Fig. 4. High-way few-shot classification accuracy on miniImageNet

Table IV displays the experimental results. Compared with other methods, although the number of parameters and memory usage of TDLPN are relatively high, its FLOPs are significantly lower than those of DN4, indicating that TDLPN still has an advantage in computational efficiency despite its increased complexity.

TABLE IV
MODEL COMPLEXITY ANALYSIS

Method	Params /10 ⁶	FLOPs /10 ⁹	Memory/GB
GNN	1.62	0.19	1.3
DN4	0.12	8.89	3.4
DPGN	5.79	1.41	1.8
TDLPN (ours)	16.11	2.88	7.74

D. Ablation experiments

1) *Impact of Different Modules on Model Classification Performance*: To evaluate the impact of LPA and GAT fusion on the model, training is conducted after removing LPA and GAT, and the experimental results are shown in Table V, demonstrating the effect of LPA and GAT fusion on the classification performance. It can be observed that there is a slight decrease in performance after removing LPA, whereas the removal of GAT has a relatively greater impact, especially in the 5-way 1-shot setting. Specifically, on the miniImageNet dataset, the 5-way 1-shot accuracy after removing LPA is 52.31%, while the accuracy after removing GAT is 52.21%, showing only a small difference. However, the 5-way 5-shot accuracy after removing GAT drops significantly to 67.03%, while it remains at 68.18% after removing LPA.

A similar trend is observed on the tieredImageNet and CUB-200-2011 datasets. The 1-shot and 5-shot accuracies after removing LPA are 55.23% and 70.43%, respectively, while the accuracies after removing GAT are 42.37% and 62.54%, respectively. The findings of this experiment demonstrate that the incorporation of GAT enhances the overall efficacy of the model, particularly in the context of the tieredImageNet dataset. The elimination of GAT has a substantial impact on classification accuracy.

TABLE V
ACCURACY OF GAT AND LPA MODULE FOR 5-WAY 1-SHOT AND 5-WAY 5-SHOT TASKS ON MINIIMAGE NET AND TIEREDIMAGE NET AND CUB-200-2011 DATASETS

Method	miniImageNet		tieredImageNet		CUB-200-2011		
	GAT	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
✓		52.31±n/a	68.18±n/a	55.23±n/a	70.43±n/a	59.36±0.16	75.67±0.25
	✓	52.21±0.20	67.03±0.17	42.37±0.20	62.54±0.19	61.54±0.42	77.23±0.39
✓	✓	57.34±0.25	74.28±0.22	59.66±0.13	76.28±0.21	65.35±0.56	81.47±0.51

TABLE VI
ACCURACY OF OPTIMIZED AND UNOPTIMIZED LPA FOR 5-WAY 1-SHOT AND 5-WAY 5-SHOT TASKS ON MINIIMAGE NET AND TIEREDIMAGE NET AND CUB-200-2011 DATASETS

Method	miniImageNet		tieredImageNet		CUB-200-2011	
	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
Unoptimized LPA	56.80±0.27	72.14±0.19	57.92±0.11	74.64±0.16	64.98±0.17	79.24±0.31
Optimized LPA	57.34±0.25	74.28±0.22	59.66±0.13	76.28±0.21	65.35±0.56	81.47±0.51

These results demonstrate that the synergistic effect of both GAT and LPA in TDLPN is crucial for the few-shot learning task, and together they improve the model's classification performance across different datasets.

2) *Effect of Optimizing the Label Propagation Algorithm:* More ablation tests will be conducted on the label propagation algorithm to analyze the impact of incorporating a momentum-optimized label propagation algorithm versus an unoptimized one. The experimental results are shown in Table VI, where the accuracy of the unoptimized label propagation algorithm is 56.80% in the 5-way 1-shot task on the miniImageNet dataset, whereas the accuracy with momentum optimization improves to 57.34%. Similarly, in the 5-way 5-shot task, the accuracy of the unoptimized LPA was 72.14%, while the optimized LPA improves to 74.28%. The optimized LPA demonstrates enhanced performance in both the 1-shot and 5-shot tasks. The optimized LPA on the tieredImageNet and CUB-200-2011 datasets also exhibits improved performance, indicating that the momentum-based optimization method can significantly enhance classification accuracy in cases of label sparsity. This outcome further demonstrates how well the optimization technique works to enhance the model's capacity to adjust to modifications in the graph topology.

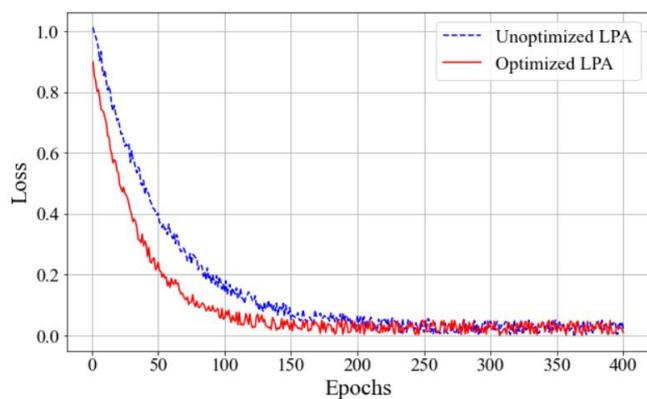


Fig. 5. Loss convergence plot for training 400 epochs on miniImageNet dataset

The momentum-based LPA significantly accelerates loss convergence and suppresses fluctuations during training, compared to the vanilla LPA. As shown in Fig. 5, the optimized LPA enables faster and smoother model convergence

by introducing a momentum term, which stabilizes the label update process and reduces oscillations. This enhancement improves the overall efficiency and stability of the label propagation algorithm.

V. CONCLUSION

This study explores the challenges faced by deep convolutional neural networks, which require a lot of labeled data to train in a resource-constrained labeling environment. In order to tackle this issue, the proposed TDLPN model integrates the label propagation algorithm and the graph attention mechanism to accurately construct a graph structure that reflects the interrelationships among input samples by optimizing the attention intensity parameter. In this way, it enables efficient label prediction. The TDLPN model outperforms the single-label propagation method on the miniImageNet, tieredImageNet, and CUB-200-2011 datasets, as demonstrated by experiments. This result not only confirms the applicability and robustness of the proposed method across different datasets but also highlights its excellent generalization ability in handling diverse few-shot classification tasks. The TDLPN model provides an effective solution for applying deep learning models in label resource-constrained conditions, proving the enormous potential of attention mechanisms and graph structures in few-shot learning.

REFERENCES

- [1] J. Bharadiya, "Convolutional neural networks for image classification," *International Journal of Innovative Science and Research Technology*, vol. 8, no. 5, pp. 673–677, 2023.
- [2] L. W. Y. G. Wang and Y. X. Geng, "Research on medical image classification based on triple fusion attention," *Engineering Letters*, vol. 33, no. 1, pp. 124–131, 2025.
- [3] R. Y. Lateef F, "Survey on semantic segmentation using deep learning techniques," *Neurocomputing*, vol. 338, pp. 321–348, 2019.
- [4] Y. L. H. Z. B. T. Chunguang Wang, Jiahao Su and F. Tian, "Research on ir gas sensor variable pressure environment compensation and monitoring analysis system based on ols-wts algorithm," *Engineering Letters*, vol. 33, no. 1, pp. 69–79, 2025.
- [5] C. Zhang, Y. Zhang, L. Zhang *et al.*, "Weakly supervised few-shot object detection with detr," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024, pp. 7015–7023.
- [6] Z. W. J. L. Yu Zhang, Ming Ma and Y. Sun, "Pod-yolo object detection model based on bi-directional dynamic cross-level pyramid network," *Engineering Letters*, vol. 32, no. 5, pp. 995–1003, 2024.
- [7] T. Ghandi, H. Pourreza, and H. Mahyar, "Deep learning approaches on image captioning: A review," *ACM Computing Surveys*, vol. 56, no. 3, pp. 1–39, 2023.

- [8] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, pp. 539–546.
- [9] O. Vinyals, C. Blundell, T. Lillicrap *et al.*, "Matching networks for one shot learning," *Advances in Neural Information Processing Systems*, vol. 29, pp. 3637–3645, 2016.
- [10] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 30, pp. 4080–4090, 2017.
- [11] H. Zhu and P. Koniusz, "Transductive few-shot learning with prototype-based label propagation by iterative graph refinement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 23 996–24 006.
- [12] Y. Luo, Z. Huang, Z. Zhang *et al.*, "Learning from the past: Continual meta-learning via bayesian graph modeling," *arXiv preprint arXiv:1911.04695*, 2019.
- [13] L. Yang, L. Li, Z. Zhang *et al.*, "Dpqn: Distribution propagation graph network for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 390–13 399.
- [14] Y. Liu, J. Lee, M. Park *et al.*, "Learning to propagate labels: Transductive propagation network for few-shot learning," *arXiv preprint arXiv:1805.10002*, 2018.
- [15] F. Sung, Y. Yang, L. Zhang *et al.*, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.
- [16] V. G. Satorras and J. B. Estrach, "Few-shot learning with graph neural networks," in *International Conference on Learning Representations*, 2018.
- [17] S. Gidaris and N. Komodakis, "Generating classification weights with gnn denoising autoencoders for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 21–30.
- [18] P. Rodríguez, I. Laradji, A. Drouin *et al.*, "Embedding propagation: Smoother manifold for few-shot classification," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI*, 2020, pp. 121–138.
- [19] H. J. Ye, H. Hu, D. C. Zhan *et al.*, "Few-shot learning via embedding adaptation with set-to-set functions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8808–8817.
- [20] X. Li, Q. Sun, Y. Liu *et al.*, "Learning to self-train for semi-supervised few-shot classification," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [21] E. Triantafillou, H. Larochelle, J. Snell *et al.*, "Meta-learning for semi-supervised few-shot classification," in *International Conference on Learning Representations*, 2018, pp. 10 276–10 286.
- [22] C. Simon, P. Koniusz, R. Nock *et al.*, "Adaptive subspaces for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4136–4145.
- [23] K. Saito, D. Kim, S. Sclaroff *et al.*, "Semi-supervised domain adaptation via minimax entropy," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8050–8058.
- [24] J. Kim, T. Kim, S. Kim *et al.*, "Edge-labeling graph neural network for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11–20.
- [25] J. Liu, L. Song, and Y. Qin, "Prototype rectification for few-shot learning," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*, 2020, pp. 741–756.
- [26] K. Ji, J. Yang, and Y. Liang, "Bilevel optimization: Convergence analysis and enhanced design," in *International Conference on Machine Learning*, 2021, pp. 4882–4892.
- [27] A. Vaswani, N. Shazeer, N. Parmar *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [28] T. Yu, S. He, Y. Z. Song *et al.*, "Hybrid graph neural networks for few-shot learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, pp. 3179–3187.
- [29] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*, 2017, pp. 1126–1135.
- [30] R. Zhang, T. Che, Z. Ghahramani *et al.*, "Metagan: An adversarial approach to few-shot learning," *Advances in Neural Information Processing Systems*, vol. 31, pp. 2365–2374, 2018.
- [31] N. Mishra, M. Rohaninejad, X. Chen *et al.*, "A simple neural attentive meta-learner," *arXiv preprint arXiv:1707.03141*, 2017.
- [32] Q. Sun, Y. Liu, T. S. Chua *et al.*, "Meta-transfer learning for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 403–412.
- [33] S. W. Yoon, J. Seo, and J. Moon, "Tapnet: Neural network augmented with task-adaptive projection for few-shot learning," in *International Conference on Machine Learning*, 2019, pp. 7115–7123.
- [34] W. Y. Chen, Y. C. Liu, Z. Kira *et al.*, "A closer look at few-shot classification," *arXiv preprint arXiv:1904.04232*, 2019.
- [35] H. J. Ye, H. Hu, D. C. Zhan *et al.*, "Learning embedding adaptation for few-shot learning," *arXiv preprint arXiv:1812.03664*, vol. 7, 2018.
- [36] Y. Liu, B. Schiele, and Q. Sun, "An ensemble of epoch-wise empirical bayes for few-shot learning," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI*, 2020, pp. 404–421.
- [37] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 657–10 665.